Probabilistic Programming Languages (PPL)

MPRI 2.40 - 2024/2025

Christine Tasson

Septembre, the 26th

Today's schedule

Semantics of Discrete Probabilistic Programming

- Why semantics ?
- Imperative language
- Functional language

Practical session

- Modeling with https://github.com/gbdrt/mu-ppl
- Computing semantics
- Limits of exact inference

Graphical Models

- Bayesian Network into μPPL
- Inference is NP-Hard
- Variable elimination heuristics

Discrete semantics

Why semantics?

What is semantics ?

Denotational semantics

Programs denote functions that act on memory (imperative language) or on values (functional language).

Christopher Strachey (1916-1975) *Towards a formal semantics (1964)*

Dana Scott (1932–) and Christopher Strachey *Towards a mathematical Semantics of Computer Languages (1971)*

Operational semantics

Transition system whose states represent memory (Turing machine, stack automata, Krivin machine,. . .) or on terms (rewriting systems,. . .)

Peter Landin (1930-2009) *The Mechanical Evaluation of Expressions (1966)*

Probabilistic Semantics

Probabilistic Programs

- Describe statistical models
- Compute or approximate the law of a random variable (and of its characteristics)

Formal methods

- How to prove that the inference of a probabilistic program denotes the right statistical model ?
- How to prove that an inference algorithm is correctly implemented ?
- How to prove correction of program transformations necessary to implement inference programs ?

Dexter Kozen, *Semantics of probabilistic programs, 1979*

Discrete semantics

Probabilistic Imperative language

Syntax of pIMP a simple subset of mu-PPL

$$t ::= \mathrm{None} \mid \mathrm{bool} \mid \mathrm{int} \mid t \times t \mid \mathrm{dist}(t)$$

$$e ::= c \mid x \mid (e,\ e) \mid op(e)$$

$$s ::= \mathrm{pass} \mid x = e \mid x = f(e) \mid \mathrm{if}\ e\colon\ s\ \mathrm{else}\colon\ s \mid s;\ s \mid x = \mathrm{sample}(e) \mid \mathrm{assume}(e)$$

$$d ::= \mathrm{def}\ f(x)\colon\ s\ \mathrm{return}\ e \mid d\ d$$

Type semantics

$$
\begin{array}{lcll}
[\![t]\!] & = & t & \textit{Countable set} \\[4pt]
[\![\mathrm{None}]\!] & = & \{*\} & \textit{one point set} \\
[\![\mathrm{bool}]\!] & = & \mathbb{B} & \textit{two points } \mathrm{True} \textit{ et } \mathrm{False} \textit{ sets} \\
[\![\mathrm{int}]\!] & = & \mathbb{N} & \textit{integers} \\
[\![t_1 \times t_2]\!] & = & t_1 \times t_2 & \textit{products} \\
\left[\!\left[\mathrm{dist}(t)\right]\!\right] & = & t \to \mathbb{R}^+ & \textit{value to score tables}
\end{array}
$$

Expression Semantics

An **environnement** Γ maps typed variables to well-typed values.

A (deterministic) **expression** $\Gamma \vdash e : t$ computed in an environnement Γ evaluates to a value in t.

It is interpreted by a set function

$$[\![e]\!] : \Gamma \to t$$

$$\begin{array}{lcl}
[\![c]\!]_\gamma^\varphi & = & c \\
[\![x]\!]_\gamma^\varphi & = & \gamma(x) \\
\left[\!\left[op(e)\right]\!\right]_\gamma^\varphi & = & op([\![e]\!]_\gamma^\varphi) \\
\left[\!\left[(e_1,\ e_2)\right]\!\right]_\gamma^\varphi & = & ([\![e_1]\!]_\gamma^\varphi, [\![e_2]\!]_\gamma^\varphi)
\end{array}$$

Commands Semantics

A **command** is a Markov process on the state of the memory represented by environments Γ.

A **stocahstic matrix** $\theta : X \rightsquigarrow Y$ is a positive real matrix $\theta \subseteq X \times Y \rightarrow \mathbb{R}^+$, is a measure on Y parameterized by X.

$\theta : X \times Y \rightarrow \mathbb{R}^+$

for all $x \in X$, $\theta_x : Y \rightarrow \mathbb{R}^+$ is a subprobability on Y $(\forall x \sum_y \theta_x(y) \leq 1)$

A **command** s transforms environments into subprobability distribution of next environments.

It is interpreted as a stochastic matrix

$$[\![s]\!] : \Gamma \rightsquigarrow \Gamma$$

Dalqvist, Kozen, Silva, Semantics of Probabilistic Programming: A Gentle Introduction, in Foundations of Probabilistic Programming, 2020

Interpretation of selected commands

$$[\![s]\!] : \Gamma \rightsquigarrow \Gamma$$

$$
\begin{array}{lcl}
[\![\mathrm{pass}]\!]^{\varphi}_{\gamma}(\gamma') & = & \delta_{\gamma}(\gamma') = 1 \text{ if } \gamma = \gamma\text{'' and 0 otherwise} \\
[\![x = e]\!]^{\varphi}_{\gamma}(\gamma') & = & \delta_{\gamma + \left[x \leftarrow [\![e]\!]^{\varphi}_{\gamma}\right]}(\gamma') \\
[\![\mathrm{if}\ e\text{: } s_1\ \mathrm{else}\text{: } s_2]\!]^{\varphi}_{\gamma} i(\gamma') & = & [\![s_1]\!]^{\varphi}_{\gamma}(\gamma')\ \mathit{if}[\![e]\!]^{\varphi}_{\gamma}\ \mathit{otherwise}[\![s_2]\!]^{\varphi}_{\gamma}(\gamma') \\
[\![x = \mathrm{sample}(e)]\!]^{\varphi}_{\gamma}(\gamma') & = & \sum_{v} [\![e]\!]^{\varphi}_{\gamma}(v)\ \delta_{\gamma + [x \leftarrow v]}(\gamma') \\
[\![x = f(e)]\!]^{\varphi}_{\gamma}(\gamma') & = & \sum_{v} \mu(v)\ \delta_{\gamma + [x \leftarrow v]}(\gamma') \mathit{with}\ \mu = \varphi(f)([\![e]\!]^{\varphi}_{\gamma}) \\
[\![\mathrm{assume}(e)]\!]^{\varphi}_{\gamma}(\gamma') & = & \delta_{\mathrm{True}}([\![e]\!]^{\varphi}_{\gamma})\ \delta_{\gamma}(\gamma') \\
[\![s_1;\ s_2]\!]^{\varphi}_{\gamma}(\gamma') & = & \sum_{\gamma_1} [\![s_1]\!]^{\varphi}_{\gamma}(\gamma_1)\ [\![s_2]\!]^{\varphi}_{\gamma_1}(\gamma')
\end{array}
$$

Declaration and Inference

$$
\begin{aligned}
[\![\text{def } f(x)\text{: } s \text{ return } e]\!]^{\varphi} &= \varphi + [f \leftarrow \lambda v.\ \lambda v'.\ \theta(v, v')] \\
&\quad \textit{with } \theta(v, v') = \sum_{\gamma} [\![s]\!]^{\varphi}_{[x \leftarrow v]}(\gamma)\ \delta_{[\![e]\!]^{\varphi}_{\gamma}}(v') \\
[\![d_1\ \ d_2]\!]^{\varphi} &= [\![d_2]\!]^{\varphi_1}\ \ \textit{with } \varphi_1 = [\![d_1]\!]^{\varphi}
\end{aligned}
$$

$$
\begin{aligned}
[\![\text{infer}(f,\ e)]\!]^{\varphi}_{\gamma}(v) &= \begin{cases} \dfrac{\mu(v)}{\mu(\top)} & \textit{if } 0 < \mu(\top) < \infty \\ \text{Error} & \textit{otherwise} \end{cases} \\
&\quad \textit{with } \mu(v) = \varphi(f)([\![e]\!]^{\varphi}_{\gamma}) \\
&\quad \textit{and } \mu(\top) = \textstyle\sum_{v} \mu(\top)
\end{aligned}
$$

Discrete semantics

Functional language

A Call-by-value probabilistic λ-calculus

Ground types: $b ::= \mathrm{unit} \mid \mathrm{bool} \mid \mathrm{int} \mid b \times b$

Types: $t ::= b\ \mathrm{dist} \mid t \otimes t \mid t \to t$

Constants: $c ::= () \mid \mathrm{true} \mid \mathrm{false} \mid \mathrm{n} \mid (c, c) \mid f(c, \ldots, c)$ where f is a ground operator

Values: $v ::= \mathrm{Dirac}\ c \mid \mathrm{Bernoulli\ r} \mid \mathrm{Dice\ n} \mid \lambda x.e \mid \mathrm{fix} x.e$

Terms: $e ::= v \mid x \mid (e, e) \mid (v)w \mid$ if v: e else: e | let $x =$ Sample e in e

Contexts: $G ::= 1 \mid G, x : t$ where $x \notin G_0$

$\boxed{G \vdash e : t}$ Type System

$$\frac{}{G \vdash () : \text{unit}} \qquad \frac{}{G \vdash \underline{\texttt{true}} : \text{bool}} \qquad \frac{}{G \vdash \underline{\texttt{false}} : \text{bool}} \qquad \frac{n \in \mathbb{N}}{G \vdash \underline{n} : \text{int}} \qquad \frac{f \in b_1 \times \cdots \times b_n \to b'}{G \vdash f(\vec{c}) : b'}$$

$$\frac{r \in \mathbb{R}}{G \vdash \text{Bernoulli } \ r : \text{bool dist}} \qquad \frac{n \in \mathbb{N}}{G \vdash \text{Dice } \ n : \text{int dist}} \qquad \frac{G \vdash c : b}{G \vdash \text{Dirac}(c) : b \text{ dist}}$$

$$\frac{G(x) = t}{G \vdash x : t} \qquad \frac{G \vdash e_1 : t_1 \qquad G \vdash e_2 : t_2}{G \vdash (e_1, \ e_2) : t_1 \otimes t_2}$$

$$\frac{G + [x : t_1] \vdash e : t_2}{G \vdash \lambda x.e : t_1 \to t_2} \qquad \frac{G \vdash e_2 : t_1 \to t_2 \qquad G \vdash e_1 : t_1}{G \vdash (e_2) e_1 : t_2} \qquad \frac{G \vdash e_1 : b \text{ dist} \qquad G + [x : b] \vdash e_2 : t_2}{G \vdash \text{let } x = \ \text{Sample } e_1 \ \text{ in } \ e_2 : t_2}$$

Operational Semantics (discrete case) Labeled Transition System (LTS)

Ugo Dal Lago, On Probabilistic Lambda-Calculi, in Foundations of Probabilistic Programming, 2020

$$\frac{}{\text{sample}(\text{Bernoulli } r) \xrightarrow{r} \underline{\texttt{true}}} \qquad \frac{}{\text{sample}(\text{Bernoulli } r) \xrightarrow{1-r} \underline{\texttt{false}}}$$

$$\frac{\forall i \in \{1, \dots, n\}}{\text{sample}(\text{Dice } n) \xrightarrow{\frac{1}{n}} \underline{i}} \qquad \frac{}{\text{sample}(\text{Dirac } c) \xrightarrow{1} c}$$

$$\frac{}{\text{let } x = c \text{ in } e_2 \xrightarrow{1} e_2[\text{Dirac c}/x]} \qquad \frac{e_1 \xrightarrow{r} e_1'}{\text{let } x = e_1 \text{ in } e_2 \xrightarrow{r} \text{let } x = e_1' \text{ in } e_2}$$

$$\frac{}{(\lambda x.e_2)v \xrightarrow{1} e_2[v/x]} \qquad \frac{e_2 \xrightarrow{r} e_2'}{(e_2)e_1 \xrightarrow{r} (e_2')e_1} \qquad \frac{e_1 \xrightarrow{r} e_1'}{(e_2)e_1 \xrightarrow{r} (e_2)e_1'}$$

$$\mathbf{Proba}(e, e') = \begin{cases} r & \text{if } e \xrightarrow{r} e' \\ 1 & \text{if } e = e' \text{ are values} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{Proba}^n(e, e') = \sum_{e_1} \mathbf{Proba}^{n-1}(e, e_1)\,\mathbf{Proba}(e_1, e')$$

$$\mathbf{Proba}^\infty(e, e') = \sup \mathbf{Proba}^n(e, e')$$

Denotational semantics of Higher Order

Discrete Probabilistic Programming

Model of Linear Logic

SMCC - Symmetric Monoidal Closed Category

Category : $\mathscr{C}$ objects + morphisms, identity and associative composition

Symmetric Monoidal : $1, A \otimes B$ + unit, associative and commutative laws

Closed: $A \multimap B, \Lambda, \mathrm{ev}$ + currying and evaluation are in tightly related, $\dfrac{A \otimes B \to C}{A \to B \multimap C}$.

Comonad

$! : \mathscr{C} \to \mathscr{C}$ a functor (action on objects and morphisms)

counit $\epsilon_A : !A \to A$ and comultiplication $\delta_A : !A \to !!A$ + diagrams.

Monoidal Strength:

natural isomorphisms $!T \xrightarrow{\sim} 1$ and $!(A \times B) \xrightarrow{\sim} !A \otimes !B$ + coherence diagrams

Commutative Comonoid

weakening $w_A : !A \to 1$ and contraction $c_A : !A \to !A \otimes !A$ + coherence diagrams

Call-By-Value in models of LL

Interpretation of types

Ground types: $Z^* = !Z$

Function types: $(A \to B)^* = !(A^* \multimap B^*)$

Types A^* are preceded by !, thus

$$A^* \xrightarrow{\delta} !A^*$$

$$A^* \xrightarrow{c} A^* \otimes A^* \qquad A^* \xrightarrow{w} 1$$

Contexts: $(A_1, \ldots, A_n)^* = {A_1}^* \otimes \cdots \otimes {A_n}^*$

Interpretation of terms: $G \vdash e : A$ is interpreted as a morphism $e^* : G^* \to A^*$

See call-by-name, call-by-value, call-by-need, and the Linear Lambda Calculus, Maraist & al.

Semantics of CBV in LL

Interpretation of terms: $G \vdash e : A$ is interpreted as a morphism $e^* : G^* \to A^*$

Variable:

$$G, x : t \vdash x : t \qquad\qquad G^* \otimes t^* \xrightarrow{w_{G^*} \otimes \mathrm{id}} 1 \otimes t^* \xrightarrow{\sim} t^*$$

Abstraction

$$\frac{G, x : t_1 \vdash e : t_2}{G \vdash \lambda x.e : t_1 \to t_2} \qquad\qquad \frac{G^* \otimes {t_1}^* \xrightarrow{e^*} {t_2}^*}{G^* \xrightarrow{\Lambda e^*} {t_1}^* \multimap {t_2}^*}$$

$$G^* \xrightarrow{\delta} !G^* \xrightarrow{!\Lambda e^*} !({t_1}^* \multimap {t_2}^*) = {t_1 \to t_2}^*$$

Application

$$\frac{G \vdash e_2 : t_1 \to t_2 \qquad G \vdash e_1 : t_1}{G \vdash (e_2)e_1 : t_1} \qquad\qquad G^* \xrightarrow{e_2} !({t_1}^* \multimap {t_2}^*) \xrightarrow{\epsilon} {t_1}^* \multimap {t_2}^* \qquad G^* \xrightarrow{e_1} {t_1}^*$$

$$G^* \xrightarrow{c} G^* \otimes G^* \to ({t_1}^* \multimap {t_2}^*) \otimes {t_1}^* \xrightarrow{ev} {t_2}^*$$

Probabilistic Coherence Spaces

A model of LL for discrete probability

Vincent Danos, Thomas Ehrhard, On probabilistic coherence spaces, 2011

PCOH

The category of Probabilistic Coherent spaces - Pcoh

Object: $(|A|, \mathrm{P}(A))$ with $|A|$ a set and $\mathrm{P}(A) \subset (\mathbb{R}^+)^{|A|}$ such that

$$\mathrm{P}(A) = \mathrm{P}(A)^{\perp\perp} \text{ where } \mathrm{P}^\perp = \{x \in (\mathbb{R}^+)^{|A|} \mid \forall x' \in \mathrm{P}\ \langle x, x' \rangle = \sum_{a \in |A|} x_a x'a \leq 1\}$$

Bounded covering $\forall a \in |A|\ (\exists x \in \mathrm{P}(A),\ x_a \neq 0)$ and $(\exists p \in \mathbb{R}^+,\ \forall x\ x_a < p)$

Examples $[\![\tau]\!] = (|\tau|, \mathrm{P}(\tau))$

$$|\texttt{unit dist}| = \{*\} \qquad \mathrm{P}(\texttt{unit dist}) = [0,1]$$

$$|\texttt{int dist}| = \mathbb{N} \qquad \mathrm{P}(\texttt{int dist}) = \left\{(x_n) \mid \sum x_n \leq 1\right\}$$

$$|\texttt{bool dist}| = \{\texttt{t},\ \texttt{f}\} \qquad \mathrm{P}(\texttt{bool dist}) = \left\{(x_\texttt{t}, x_\texttt{f}) \mid x_\texttt{t} + x_\texttt{f} \leq 1\right\}$$

$$|A \times B| = |A| \uplus |B| \qquad \mathrm{P}(A \times B) = \{(x_i)_{i \in |A| \uplus |B|} \mid (x_i)_{i \in |A|} \in \mathrm{P}(A), (x_i)_{i \in |B|} \in \mathrm{P}(B)\}$$

$$[\![\text{Bernoulli p}]\!] = (p, 1-p) \qquad [\![\text{Dice n}]\!] = (\tfrac{1}{n}, \ldots, \tfrac{1}{n}, 0, \ldots, 0, \ldots)$$

PCOH and Linear coherent maps is a model of simply typed lambda-calculus

The linear category of Probabilistic Coherence Spaces

Object: $(|A|, \mathrm{P}(A))$ with $|A|$ a set and $\mathrm{P}(A) \subset (\mathbb{R}^+)^{|A|}$ set of functions

Morphism: $f : (|A|, \mathrm{P}(A)) \to (|B|, \mathrm{P}(B))$ a matrix $(f_{(a,b)})$ indexed by $|A| \times |B|$ such that

$$\forall x \in \mathrm{P}(A), f(x) : b \mapsto \sum_{a \in |A|} x(a) f_{a,b} \in \mathrm{P}(B)$$

Examples

$$f : [\![\text{bool dist}]\!] \to [\![\text{bool dist}]\!] \text{ such that } f = \begin{array}{c c} W \backslash S & \begin{array}{cc} \mathrm{T} & \mathrm{F} \end{array} \\ \begin{array}{c} \mathrm{T} \\ \mathrm{F} \end{array} & \begin{bmatrix} 1/5 & 4/5 \\ 3/4 & 1/4 \end{bmatrix} \end{array}$$

PCOH and Linear coherent maps is a model of Linear Logic

The linear category of Probabilistic Coherence Spaces

Object: $(|A|, \mathrm{P}(A))$ with $|A|$ a set and $\mathrm{P}(A) \subset (\mathbb{R}^+)^{|A|}$ set of functions
Morphism: $f : (|A|, \mathrm{P}(A)) \to (|B|, \mathrm{P}(B))$ such that $f \cdot \mathrm{P}(A) \subseteq \mathrm{P}(B)$

Tensor product

$|X \otimes Y| = |X| \times |Y|$
$\mathrm{P}(X \otimes Y) = \{x \otimes y \mid x \in \mathrm{P}(X),\ y \in \mathrm{P}(Y)\}^{\perp\perp}$ where $(x \otimes y) : (a, b) \mapsto x(a)y(b)$

Examples of morphisms in PCOH:

Duplication: $\Delta : [\![\text{b dist}]\!] \to [\![b \times b\text{dist}]\!]$ such that $\Delta(x) = \sum_a x_a \delta_{a,a}$
Marginalization: $\text{proj} : [\![\text{b dist}]\!] \otimes [\![\text{b' dist}]\!] \to [\![\text{b dist}]\!]$ such that $\text{proj}(x \otimes y) = x$

Exponential

$|!X| = \mathscr{M}_{\text{fin}}(|X|)$
$\mathrm{P}(!X) = \{x^! \mid x \in \mathrm{P}(X)\}^{\perp\perp}$ where $x^! : m \mapsto \prod_{a \in m} x(a)^{m(a)}$

The cartesian closed category of Probabilistic Coherence spaces

The category of Probabilistic Coherence spaces and analytic maps

Object: $(|A|, \mathrm{P}(A))$ with $|A|$ a set and $\mathrm{P}(A) \subset (\mathbb{R}^+)^{|A|}$ set of functions

Morphism: $f : (|A|, \mathrm{P}(A)) \to (|B|, \mathrm{P}(B))$ a matrix $(f_{(m,b)})$ indexed by $\mathscr{M}_{\mathrm{fin}}(|A|) \times |B|$ such that

$$\forall x \in \mathrm{P}(A), f(x) : b \mapsto \sum_{m \in \mathscr{M}_{\mathrm{fin}}(|A|)} \prod_{a \in m} x(a)^{m(a)} f_{m,b} \in \mathrm{P}(B)$$

Examples

$f : [\![unit]\!] \to [\![unit]\!]$ such that $\forall x \in [0,1],\ f \cdot x = \sum_n f_n x^n \in [0,1]$

$f : [\![bool]\!] \to [\![unit]\!]$ such that $f_{(\mathtt{true}^n, *)} = 1$ otherwise $f_{m,*} = 0$, then $f \cdot (p, 1-p) = \sum_n p^n$ and $f \cdot (1,0) = 0$.

PCOH and analytic maps is a CCC (the Kleisli Category of !) CPO enriched.

It is a model of PCF = CBN lambda calculus and fixpoints

Results on Probabilistic Coherent Spaces

Compositionality

$$\llbracket (e)e_2 \rrbracket_b = \llbracket e \rrbracket(\llbracket e_2 \rrbracket)_b = \sum_m \llbracket e \rrbracket_{m,b} \prod_{a \in m} \llbracket e_2 \rrbracket_a^{m(a)}$$

Invariance of the semantics

$$\llbracket e \rrbracket = \sum_{e_2} \mathbf{Proba}(e, e_2) \llbracket e_2 \rrbracket$$

Adequacy Lemma

$$\text{if } \vdash e : \mathtt{nat}, \text{ then } \mathbf{Proba}^{\infty}(e, \underline{n}) = \llbracket e \rrbracket_n$$

Full Abstraction at ground type nat

$$\llbracket e \rrbracket_1 = \llbracket e \rrbracket_2 \text{ if and only if } \mathbf{Proba}^{\infty}(C[e_1], n) \overset{\forall C[] \forall n}{=} \mathbf{Proba}^{\infty}(C[e_2], n)$$

Why you should care or not on Probabilistic Coherence Spaces

✓ Interpretation of probabilistic programs of discrete type (int or real) are PCOH maps, with good analytic Properties

Probabilistic Stable Functions on Discrete Cones are Power Series. R. Crubille, 2018

✓ Concrete Sandbox for getting intuitions on probabilistic programs

Non definability: $\mathrm{Pcoh}(\mathrm{bool}, 1) = \left\{ Q \in (\mathbb{R}^+)^{\mathcal{M}_{\mathrm{fin}}(\mathtt{t},\mathtt{f})} \mid Q_{[\mathtt{t}^n, \mathtt{f}^m]} \leq \frac{(n+m)^{n+m}}{n^n m^m} \right\}$ but the greatest we can get is $\{\!|e|\!\} \leq \frac{(n+m)!}{n!m!}$

fix fun x (*$\to$*) if x then if x then f(x) else () else if x then () else f(x)

Why you should care or not on Probabilistic Coherence Spaces

- It is not computable and thus cannot be used to implement inference
  if $\vdash^P M : \tau$ and $\{\!|M|\!\} \in \mathrm{P}(\tau)$ then $\vdash^D \text{infer } M : \tau\, dist$ and $(\!|\text{infer } M|\!)$ is a subprobability distribution over τ.

$$(\!|\text{infer } M|\!) = \frac{\{\!|M|\!\}}{\sum_{a \in |\tau|} \{\!|M|\!\}_a}$$

Scaling exact inference for discrete probabilistic programs. Holtzen & al. 2020

Practical session

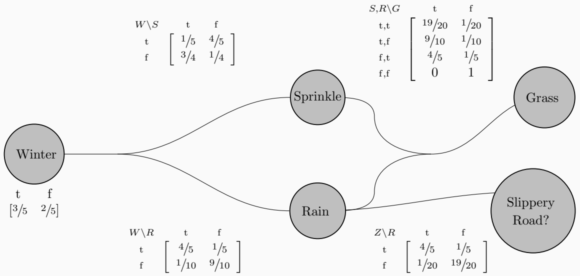
Semantics and Modeling

Graphical Models

Bayesian Networks to μ-PPL

📕 Koller & Friedman, Probabilistic Graphical Models: Principles and Techniques, Chapter 9.

Bayesian Network



Definition: A bayesian network is given by

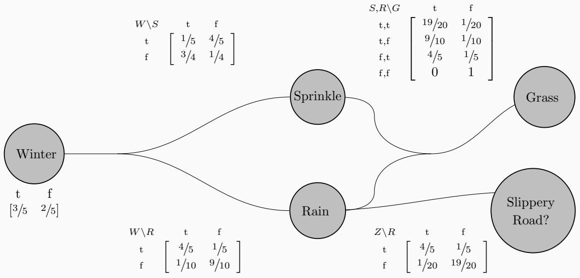
Directed Acyclic Graph (DAG) defining dependency:
The probability of X given variables depends only on parents $\mathtt{Pa}(X)$:

$$\mathbb{P}(X|\mathtt{Vars}) = \mathbb{P}(X|\mathtt{Pa}(X))$$

Conditional Probability Tables (CPT): for all R.V. X with sample set $|X|$, a matrix

$$\mathbb{P}(X|\mathtt{Pa}(X)) : \left|\mathtt{Pa}(X)\right| \times |X| \to \mathbb{R}^+$$

Bayseian Network

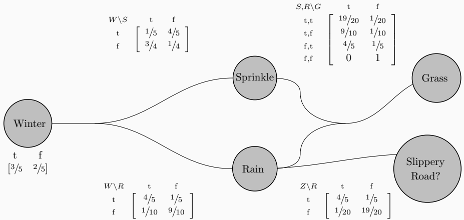


Purpose: compact representation of the joint distribution $\mathbb{P}(G, Z, S, R, W)$

What is the dimension of the mass function $\mathbb{P}(G = g, Z = z, S = s, R = r, W = w)$?

What is the dimension of the CPTs ?

Bayesian Network



Joint distribution from conditional probability tables

$$\mathbb{P}(Z,G,R,S,W) = \mathbb{P}(Z|R)\mathbb{P}(G|S,R)\mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)$$

Conditional probability

$$\mathbb{P}(Z,G,R,S,W) = \mathbb{P}(Z|G,S,R,W)\ \mathbb{P}(G,S,R,W)$$

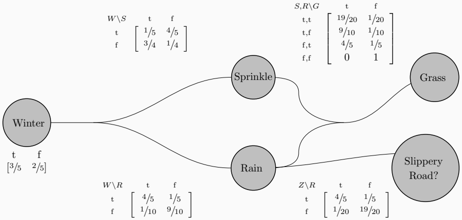
Chain Rule

$$\mathbb{P}(Z,G,R,S,W) = \mathbb{P}(Z|G,S,R,W)\mathbb{P}(G|S,R,W)\mathbb{P}(S|R,W)\mathbb{P}(R|W)\mathbb{P}(W)$$

Dependency

$$\mathbb{P}(Z|G,S,R,W) = \mathbb{P}(Z|R) \quad \mathbb{P}(G|S,R,W) = \mathbb{P}(G|S,R) \quad \mathbb{P}(S|R,W) = \mathbb{P}(S|W)$$

Bayesian Network



Querries: compute

$\mathbb{P}(G)$ with:

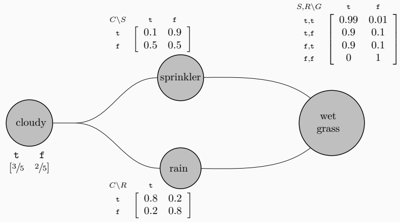
Joint distribution: $\mathbb{P}(G,R,S,W) = \mathbb{P}(G|S,R)\ \mathbb{P}(S|W)\ \mathbb{P}(R|W)\ \mathbb{P}(W)$

Marginal $\mathbb{P}(G) = \displaystyle\sum_{(r,s,w)\in|R|\times|S|\times|W|} \mathbb{P}(G,R,S,W)$

$\mathbb{P}(G,S)$, the joint distribution or $\mathbb{P}(S|G=\mathtt{t})$ the conditional distribution given evidence.

Bayesian network in μ-PPL

What is $\mathbb{P}(R|G = \mathtt{t})$?



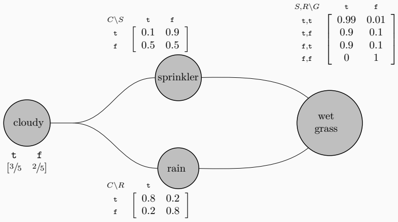

```
def grass() → bool:
    cloudy = sample(Bernoulli(0.5), name="c")
    p_rain = .8 if cloudy else 0.2
    rain = sample(Bernoulli(p_rain), name="r")
    p_sprinkler = 0.1 if cloudy else 0.5
    sprinkler = sample(Bernoulli(p_sprinkler), name="s")
    p_wet = 0.99 if (sprinkler and rain) else (0.9 if (
            sprinkler or rain) else 0)
    wet_grass = sample(Bernoulli(p_wet), name='w')
    assume(wet_grass)
    return rain

with Enumeration():
    dist: Categorical[float] = infer(grass)
```

What is the size of the enumeration tree ?

Bayesian network in μ-PPL

What is $\mathbb{P}(R|G = \mathtt{t})$?




```
def grass() → bool:
    cloudy = sample(Bernoulli(0.5), name="c")
    p_rain = .8 if cloudy else 0.2
    rain = sample(Bernoulli(p_rain), name="r")
    p_sprinkler = 0.1 if cloudy else 0.5
    sprinkler = sample(Bernoulli(p_sprinkler), name="s")
    p_wet = 0.99 if (sprinkler and rain) else (0.9 if (
            sprinkler or rain) else 0)
    wet_grass = sample(Bernoulli(p_wet), name='w')
    assume(wet_grass)
    return rain

with Enumeration():
    dist: Categorical[float] = infer(grass)
```

What is the size of the enumeration tree ? v^k where v is the maximal cardinal of the sample sets and k is the number of random variables.

Graphical Models

Inference is NP-Hard

Inference is NP-Hard

Conditional Probability Task: Given a Bayesian Network and a variable X with value set $|X|$, decide wether $\mathbb{P}(X = x) > 0$, for an $x \in |X|$.

Complexity: The conditional Probability Task is NP-Hard.

Proof:

> **The problem is NP:** given all samples, computing $\mathbb{P}(X = x)$ is linear (multiply all the concerned factors).
>
> **Reduction to 3-SAT:** given a 3-SAT formula φ, construct in polynomial time a Bayesian Network B_φ with a distinguished random variable X such that $\mathbb{P}(X = x) > 0$ if and only if φ is satisfiable.

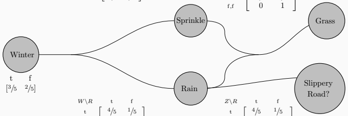
Conclusion: inference in PPL is untractable.

Graphical Models

Heuristics: Variable Elimination

Darwiche, Modeling and Reasoning with Bayesian Networks, Chapter 4 and 5.

Bayesian Network - Computing Marginals



$$W \backslash S \quad \begin{array}{c|cc} & t & f \\ t & 1/5 & 4/5 \\ f & 3/4 & 1/4 \end{array}$$

$$S,R \backslash G \quad \begin{array}{c|cc} & t & f \\ t,t & 19/20 & 1/20 \\ t,f & 9/10 & 1/10 \\ f,t & 4/5 & 1/5 \\ f,f & 0 & 1 \end{array}$$

$$W: \quad \begin{array}{cc} t & f \\ 3/5 & 2/5 \end{array}$$

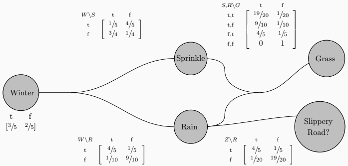
$$W \backslash R \quad \begin{array}{c|cc} & t & f \\ t & 4/5 & 1/5 \\ f & 1/10 & 9/10 \end{array}$$

$$Z \backslash R \quad \begin{array}{c|cc} & t & f \\ t & 4/5 & 1/5 \\ f & 1/20 & 19/20 \end{array}$$

Using Chain Rule naive methods

$$\begin{aligned} \mathbb{P}(Z, G) &= \sum_{w,s,r} \mathbb{P}(Z, G, S, R, W) \\ &= \sum_{w,s,r} \mathbb{P}(Z|R)\mathbb{P}(G|S,R)\mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W) \end{aligned}$$

Bayesian Network - Computing Marginals



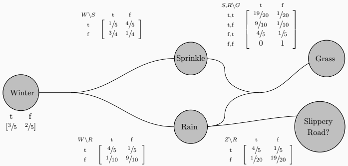
Using Chain Rule naive methods

$$\begin{aligned}\mathbb{P}(Z,G) &= \sum_{w,s,r} \mathbb{P}(Z,G,S,R,W) \\ &= \sum_{w,s,r} \mathbb{P}(Z|R)\mathbb{P}(G|S,R)\mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)\end{aligned}$$

Using Memoization and Factorization Optimized methods.

$$\mathbb{P}(Z,G) = \sum_{r} \mathbb{P}(Z|R)\left(\sum_{s} \mathbb{P}(G|S,R)\left(\sum_{w} \mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)\right)\right)$$

Bayesian Network - Computing Marginals



Using Chain Rule naive methods

$$
\begin{aligned}
\mathbb{P}(Z,G) &= \sum_{w,s,r} \mathbb{P}(Z,G,S,R,W) \\
&= \sum_{w,s,r} \mathbb{P}(Z|R)\mathbb{P}(G|S,R)\mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)
\end{aligned}
$$

$3 * 2 + 2^5 = O(v^k)$ sums and products.

Using Memoization and Factorization Optimized methods.

$$
\mathbb{P}(Z,G) = \sum_{r} \mathbb{P}(Z|R) \left(\sum_{s} \mathbb{P}(G|S,R) \left(\sum_{w} \mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W) \right) \right)
$$

$(2+2*3)+(2+2*1)+(2+2*1) = O(v^{\text{width}})$ sums and products

Algorithm 1 Variable Elimination Algorithm

input: B a Baysesian Network, $Q \subset \mathcal{V}$ query variables, π an ordering of $\mathcal{V} \setminus Q$

output: $\mathbb{P}(Q)$

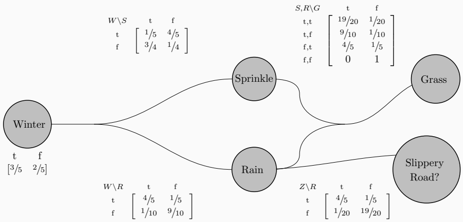
$\Phi \leftarrow$ CPTs of network
for $X \in \pi$ **do**
  for $x \in |X|$ **do**
    $f_x \leftarrow \prod_k f_k$, where $f_k \in S$ and mention variable $X = x$
  end for
  $f_X \leftarrow \sum_{x \in |X|} f_x$ and $\Phi \leftarrow f_X \cup \Phi \setminus \{f_k\}$
end for

$$\begin{aligned}
\mathbb{P}(Z,G) &= \sum_r \mathbb{P}(Z|R)\left(\sum_s \mathbb{P}(G|S,R)\left(\sum_w \mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)\right)\right) && W \\
&= \sum_r \mathbb{P}(Z|R)\left(\sum_s \mathbb{P}(G|S,R) f_W(S,R)\right) && S \\
&= \sum_r \mathbb{P}(Z|R)\ f_S(R,G) && R \\
&= f_R(Z,G)
\end{aligned}$$

Variable Elimination Evaluation

An algorithm for exact inference, based on a chosen order to eliminate variables (i.e. marginalize): in the following factorization, the order is W then S then R.

$$\mathbb{P}(Z,G) \quad = \quad \sum_r \mathbb{P}(Z|R)\left(\sum_s \mathbb{P}(G|S,R)\left(\sum_w \mathbb{P}(S|W)\mathbb{P}(R|W)\mathbb{P}(W)\right)\right)$$



Exponential blow up: factor size $O(v^k)$

(v max number of samples, max width of factors)

Different orders have different perfomances

Choosing an optimal order is known as NP-hard

Many heuristic depending on the structure of the graph

Cooper, The computational complexity of probabilistic inference using bayesian belief networks, Artificial Intelligence 42 (1990), no. 2, 393405

Conclusion

Skills and knowledge

Take home

Semantics of IMP and pPCF using Markov Processes

Inference is **intractable** (even in discrete Bayesian Networks)

Heuritics for efficient exact inference (in practical case)